

研究生课程《地面无人平台自主控制》

## 建图与重定位实验

## 实验总结报告

姓    名： popfishy

学    号： \*\*\*\*\*

日    期： 2024 年 1 月 27 日

# 一、实验目的

## 1.1 实验目标

本次实验必选作业为：利用 KITTI odometry (sequence 07) 数据，将多帧雷达数据和图像数据利用位姿真值拼接起来生成场景三维地图。完成上述任务后，从以下六个任务中任选其一：

- (1) 提取图像特征，利用对应雷达点的深度将特征点存到三维全局地图中，进而实现基于特征点匹配的图像全局定位；（可以参考 ORB-SLAM 算法）
- (2) 将构建的全局地图看成三维点地图，利用 ICP 算法实现帧到全局地图的匹配定位；
- (3) 参考 NDT 地图表示方法，将地图存成 NDT 地图形式，实现基于 NDT 的地图匹配定位；（可以参考 Autoware 中的实现）
- (4) 尝试将三维地图变成二维高程/反射率地图，将当前帧也表示成二维高度/反射率图，进而实现基于 CSM 的匹配定位；
- (5) 自己设计其它的匹配定位算法；
- (6) 尝试运行 SVO-pro 或 DSO 代码，得到半致密/致密的深度地图。实验报告中需要将代码中的关键步骤和中间结果展示出来。

本实验选择实验 2，即将构建的全局地图看成三维点地图，利用 ICP 算法实现帧到全局地图的匹配定位。

## 1.2 实验环境

- (1) Ubuntu20.04 环境 +ROS1 noetic 版本
- (2) 编程语言 C++ 和 Python
- (3) ROS 默认版本的 PCL 库、Eigen 库、OpenCV 库。Open3d 库。

# 二、实验方案、原理及方法

## 2.1 总体实验方案

本实验在 Ubuntu 系统下使用 ROS noetic 版本进行开发，使用 C++ 和 Python 语言完成代码编写。

首先，将不同时间截下的雷达数据帧转换到初始时间截下，并通过相应坐标系之间的关系，对雷达数据进行拼接操作，同时根据同一坐标下相机采集的 RGB 图像数据，对相应的点云进行染色，在进行降采样的同时，去除无效点云，生成全局彩色地图。完成必选实验任务。

在上述步骤，不进行染色，得到实验二所需的全局三维点地图。编写 ICP 算法实现任意帧到全局地图的匹配定位。

## 2.2 实验基本原理

### 2.2.1 KITTI odometry 数据集的介绍

KITTI 数据集，其是由德国卡尔斯鲁厄理工学院和丰田工业大学芝加哥分校联合赞助的用于自动驾驶领域研究的数据集。作者收集了长达 6 个小时的真实交通环境，数据集由经过校正和同步的图像、雷达扫描、高精度的 GPS 信息和 IMU 加速信息等多种模态的信息组成。作者还在数据集官网提供了光流、物体检测、深度估计等多种任务的 Benchmark。

相应的传感器配置如下：

- \* 2 个一百四十万像素的 PointGray Flea2 灰度相机；
- \* 2 个一百四十万像素的 PointGray Flea2 彩色相机；
- \* 4 个 Edmund 的光学镜片，水平视角约为  $90^\circ$ ，垂直视角约为  $35^\circ$ ；
- \* 1 个 64 线的 Velodyne 旋转激光雷达， $10\text{Hz}$ ，角分辨率率为  $0.09$  度，每秒约一百三十万个点，水平视场  $360^\circ$ ，垂直视场  $26.8^\circ$ ，至多 120 米的距离范围；
- \* 1 个 OXTS RT3003 组合导航系统，6 轴， $100\text{Hz}$ ，分辨率为  $0.02$  米， $0.1^\circ$ 。

实验车辆配置如图 1 所示，从图中可知 KITTI 数据集对应的雷达、IMU 以及相机坐标系具体定义：

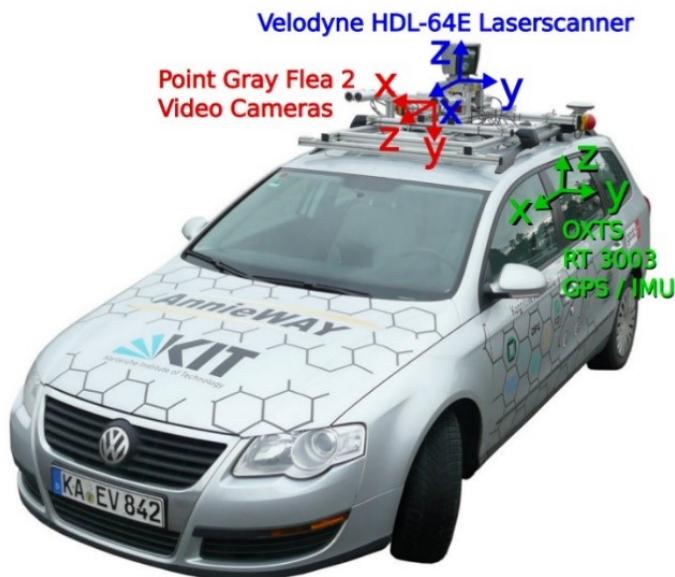


图 1 实验数据采集平台

不同传感器空间位置关系如图 2 所示，作者为我们提供了不同传感器之间坐标系变换参数。

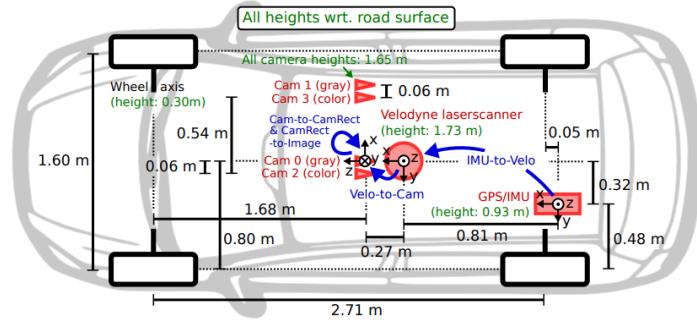


图 2 传感器空间位置关系图

### 2.2.2 单帧点云图像融合

点云图像融合本质上涉及点云的着色问题。由于激光雷达和相机具有不同的视角范围，我们可以根据相机成像模型确定相机视野内的点云，并将其着色为对应的图像像素点。其他超出相机视野的点云将被舍弃，从而完成单帧点云图像融合，之后再根据不同时刻位姿进行全局点云融合。

单目相机成像模型如图

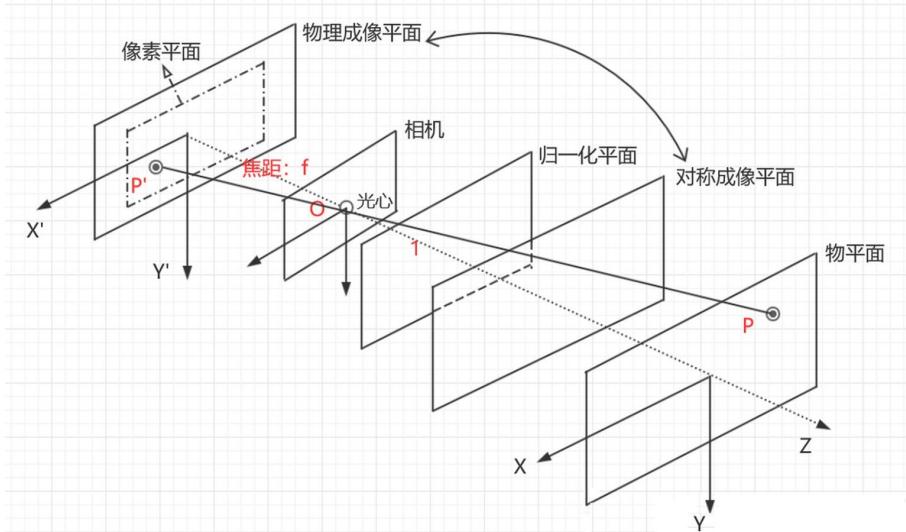


图 3 单目相机成像模型

假设  $O - XYZ$  为图像物理坐标系,  $O' - uv$  为图像像素坐标系,  $o - xyz$  为相机坐标系,  $o_w - x_w y_w z_w$  为世界坐标系。 $f$  为像距,  $L$  为物距。

由于相机坐标系可由世界坐标系平移、旋转得到, 因此, 世界坐标系中的任意一点  $p_w(x_w, y_w, z_w)$  与相机坐标系中对应点  $p(x, y, z)$  之间存在矩阵映射变换关系, 如式 (1) 所示。

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (1)$$

根据三角形相似原理，可知图像物理坐标系中对应点  $P$  的横纵坐标  $X$  和  $Y$ ，如式 (2) 所示。

$$X = f \frac{x}{L}, Y = f \frac{y}{L} \quad (2)$$

在图像像素坐标系下，点的位置指的是图像像素矩阵的行与列，设光轴在像素坐标系下的投影（也就是图像的正中心）为  $(u_0, v_0)$ ，那么不，难看出图像物理坐标系中的点  $P(X, Y)$  与图像像素坐标系中对应点  $P'(u, v)$  之间的变换关系，如式 (3) 所示。

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (3)$$

其中， $dx$ 、 $dy$  分别表示相机一个像素在  $u$ 、 $v$  方向上的真实尺寸。

综合式 (1)、(2)、(3) 可以推导出单目相机成像模型中世界坐标系到图像像素坐标系的变换关系，如 4：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{L} \begin{bmatrix} \frac{f}{dx} & 0 & u_0 & 0 \\ 0 & \frac{f}{dy} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (4)$$

激光雷达采集到的点包括三维世界的坐标，根据上述公式，得到对应图像的像素坐标，将对应像素的 RGB 值赋予点云即可完成点云图像融合。

### 2.2.3 ICP 算法原理

给定两幅点云地图  $X = \{x_1, x_2, x_3, \dots, x_{N_x}\}$ ,  $Y = \{y_1, y_2, y_3, \dots, y_{N_y}\}$  其中  $x_i$  和  $y_i$  表示该点云地图中的第  $i$  个点云的坐标， $N_x$  和  $N_y$  分别表示两幅点云地图的点云数量。我们的目标是找到一种最优的相对位姿变换  $[Tr_{x\_y}|t_{x\_y}]$ ，其可以将  $Y$  中的点云变换到  $X$  中。注意到并不是所有  $Y$  中的点都能够变换到  $X$  中。由于  $X$  和  $Y$  分别是雷达位于不同位置对同一环境的两次采样，因此必然存在某些  $Y$  中的点在  $X$  中找不到对应点。反之亦然。不失一般性，我们假设  $X$  中的  $N_x$  个点都能在  $Y$  中找到对应点，尝试极小化式 (5)。

$$E(R, t) = \frac{1}{N_x} \sum_{i=1}^{N_x} \|R_{y\_x} x_i + t_{y\_x} - y_i\|^2 \quad (5)$$

代入点云质心位置  $\mu_x$ 、 $\mu_y$  进行化简，得到式 (6)。

$$E(R, t) = \frac{1}{N_x} \sum_{i=1}^{N_x} \left\| Rx_i' - y_i' + (R\mu_x - \mu_y + t) \right\|^2 \quad (6)$$

令  $t = \mu_y - R\mu_x$ ，继续化简，得到式 (8)。

$$E(R, t) = \frac{1}{N_x} \left[ \sum_{i=1}^{N_x} \|x_i'\|^2 + \sum_{i=1}^{N_x} \|y_i'\|^2 - 2\text{tr}(RW) \right], W = \sum_{i=1}^{N_x} x_i' y_i'^T \quad (7)$$

为了极小化  $E$ , 需要取  $\text{tr}(RW)$  极大值, 记  $W = U \sum V^T$ ,  $R = VU^T$ , 因此  $RW = V \sum V^T$  是一个正定矩阵。根据引理: 对于任何正定矩阵  $AA^T$  以及任一正交阵  $B$ , 都有  $\text{tr}(AA^T) \geq \text{tr}(BAA^T)$ , 我们可知  $\text{tr}(RW) \geq \text{tr}(BRW)$  对于任意矩阵  $B$  都成立, 因此  $R$  比任  $BR$  都好。因此我们得到解, 如式 (8)。

$$\begin{cases} R = VU^T \\ t = \mu_y - R\mu_x \end{cases} \quad (8)$$

当  $x_i$ 、 $y_i$  均为正确的对应点对时, 得到正确的相对位姿。但实际使用中难以得到正确的对应点对, 因此, 常用的方法是: 根据当前估计的相对位姿, 计算  $Y$  中与  $Tr \times x_i$  最接近的点, 将其当成  $x_i$  的对应点, 经过多次迭代不断求解新的  $Tr$ , 直到  $Tr$  收敛或达到最大迭代次数。

考虑到本实验中使用的 Kitti Odometry 数据集场景较大, 纯粹的 ICP 匹配需要很长时间进行迭代, 而且可能掉入局部极值, 导致匹配错误。因此, 选择人为提供一个较优的位姿初值进行匹配。

## 三、实验过程及结果分析

### 3.1 实验过程

首先, 利用 KITTI odometry (sequence 07) 数据, 将多帧雷达数据和图像数据利用位姿真值拼接起来生成场景三维地图:

- (1) 读取位姿文件。即 sequences/07/poses.txt 文件, 该文件共有 1101 行, 对应 1101 帧位姿数据; 共 12 列, 对应  $3 \times 3$  的旋转矩阵  $R$  和  $3 \times 1$  的平移矩阵  $T$ 。
- (2) 对同一时间截下的点云和图像进行融合。KITTI odometry 数据集提供的雷达点云数据保存在 sequences/07/velodyne 目录下, 提供的图像数据保存在 sequences/07/image\_X (本实验仅使用第 4 个相机的数据, 即 image\_3) 目录下。使用 PCL 点云库将.bin 格式点云数据读入, 获取点云相对激光雷达的三维坐标, 使用 OpenCV 库读取图像数据。原.bin 数据点云格式为  $[x, y, z, \text{reflectance value}]$ , 代表点云在雷达数据采集时刻相对于雷达坐标系的三维坐标, reflectance value 表示雷达反射率(无用信息)。另外注意, 要将每一帧点云和图像转换到同一原点的坐标系下, 需要提前获取雷达和相机的位姿, 即外参。外参数数据存储在 sequences/07/calib.txt 文件中。其中前四行为相机外参矩阵 (第 4 个相机外参矩阵为  $P3$ ), 第五行为雷达外参矩阵 ( $Tr$ )。之后根据图像 RGB 像素值, 将对应点云进行着色, 并滤除超出相机视野范围的点云。
- (3) 拼接雷达图像融合数据, 生成全局地图。取 sequences/07/poses.txt 中的位姿数据, 根据每一帧时间截下的车辆姿态相对初始时间截的车辆姿态变换矩阵, 将点云数据全部变换到初始时间截下相机坐标系中。将每一帧点云根据对应时间截的位姿数据变换到初始时间下的原点坐标系中, 使用 pcl\_viewer 可视化工具显示最终结果。

利用上述任务构建的全局点云地图，使用 ICP 算法实现帧到全局地图的匹配定位：

- (1) 读取之前 Kitti odometry 数据集生成的全局点云地图。
- (2) 编写局部点云发布程序，发布任意一帧局部点云地图。
- (3) 在 Rviz 内使用 2D Pose Estimate 工具提供局部地图初始位姿，若在该位姿下进行的 ICP 匹配重叠区域较高（大于设定阈值，本实验设定为 0.75），则可认为其较优，可采纳，进入迭代循环，完成匹配。

### 3.2 实验结果分析

多帧雷达数据和图像数据生成场景三维地图实验结果见图 4。总体上说实验结果较好，彩色点云局部车辆清晰可见。



(a) 全局彩色点云

(b) 局部彩色点云

图 4 多帧雷达数据融合图像数据实验结果

利用构建的全局点云地图，使用 ICP 算法实现帧到全局地图的匹配定位。该工作时基于 FAST LIO LOCALIZATION 的源码进行修改，去除了 SLAM 建图过程，调整重定位模块数据接口和数据订阅发布方式，编写了局部点云发布程序。载入全局点云地图，发布局部单帧点云数据，运行 ICP 重定位程序，使用 2D Pose Estimate 工具提供初始位姿。ICP 重定位程序根据提供的初始位姿，计算匹配度，若匹配度较低，匹配结果会比较粗糙，舍弃该位姿；若匹配度高，迭代后匹配效果会更好，采纳该位姿，进行迭代计算，完成匹配。实验结果见图 5。

## 四、实验心得与体会

- (1) 本次实验可以看成是《KITTI 数据集位姿转换与雷达点云数据融合生成局部地图实验》的进阶版，主要在该实验基础上，了解了点云染色的方法，熟悉了 Open3D 和 PCL 库的使用。

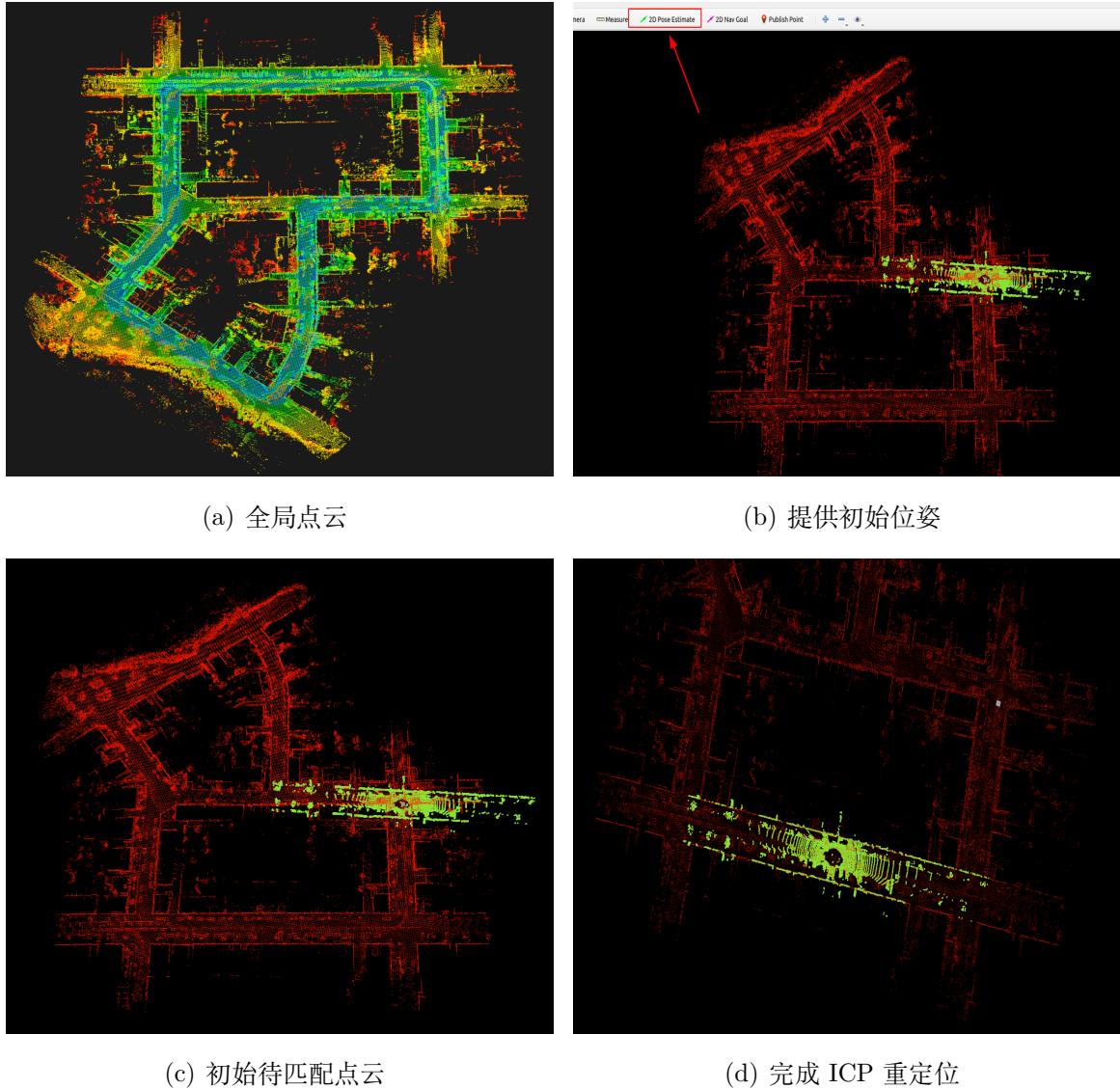


图 5 ICP 重定位实验结果

- (2) 对于 KITTI 数据集组成、各文件具体含义、不同坐标系之间转换关系都有了深入理解。
- (3) 在对 FAST LIO LOCALIZATION 解耦的过程中，加深了我对于 SLAM 算法各组成部分、各部件之间的功能和关系的理解。强化了代码编程调试能力。

## References

- [1] 点云染色：<https://github.com/SpadyDong/CloudMerger-KITTI>
- [2] ICP 重定位：[https://github.com/HViktorTsoi/FAST\\_LIO\\_LOCALIZATION](https://github.com/HViktorTsoi/FAST_LIO_LOCALIZATION)